Violin MEMORY

Best Practices

Concerto Best Practices Guide for VMware vSphere® Deployment

Document Number: 535-0251-00 Rev 02
July 2016

Table of Contents

LEGAL NOTICE

Violin Memory, Inc.
4555 Great America Parkway
Santa Clara, CA 95054
USA

Overview

This document provides best practices for deploying VMware vSphere® 5.x and newer versions on Violin 7000 Series Flash Storage Platforms (FSPs) using either a Fibre Channel or iSCSI storage area network (SAN). The document includes best practices, which are recommendations on how to set up and configure both VMware ESXi™ and the FSP for best performance, usability and reliability. While all of the items in this best practices guide are recommended, they are not necessarily required in every configuration or customer application. Most of the best practices in this guide apply to both FC and iSCSI implementations. The differences will be highlighted in the iSCSI chapter.

It is assumed that the user of this guide has basic vSphere and Violin administration knowledge.

Intended Audience

This guide is intended for IT professionals who have basic vSphere and Violin administration knowledge and experience with installing and configuring VMware ESXi for use with Fibre Channel and iSCSI storage area networks.

Reference Documents

The following documents are references used in this guide and can provide additional information on best practices and procedures for implementing VMware vSphere solutions with Violin arrays.

Violin Memory Documentation

The following documents are available from Violin Memory:

- *Violin Memory Interoperability Best Practices Guide*
- *Concerto 7000 All Flash Array Best Practices Guide*
- *SQL Server Best Practices Guide*
- *Best Practice Guide for Oracle Database*

VMware Documentation

The following documents are available from VMware:

- VMware Performance Best Practices for VMware vSphere
- vSphere Storage Guide
- VMware vSphere Configuration Maximums
- Connecting to an ESXI host using an SSH client
- Adjusting Round Robin IOPS limit from default 1000
- Changing the queue depth for Qlogic, Emulex, and Brocade HBAs
- Configuring disks to use VMware Paravirtual SCSI (PVSCSI) Adapters
- PVSCSI queue depths for large scale workloads
- VMware VAAI
- FAQ for VAAI

- Storage I/O Control Technical Overview and Considerations for Deployment
- Using esxtop to identify storage performance issues
- Interpreting esxtop Statistics
- All Paths Down timeout for a storage device has expired
- Host level performance charts display large differences in the CPU Usage and Usage in MHz metrics when compared to the CPU Demand metric
- VMware OS Optimization Tool
- Support statement for 512e and 4K Native drives for VMware vSphere
- Frequently Asked Questions on VMware vSphere 5.x for VMFS-5
- Block size limitations of a VMFS datastore
- Accessing or creating a VMFS datastore larger than 16TB fails

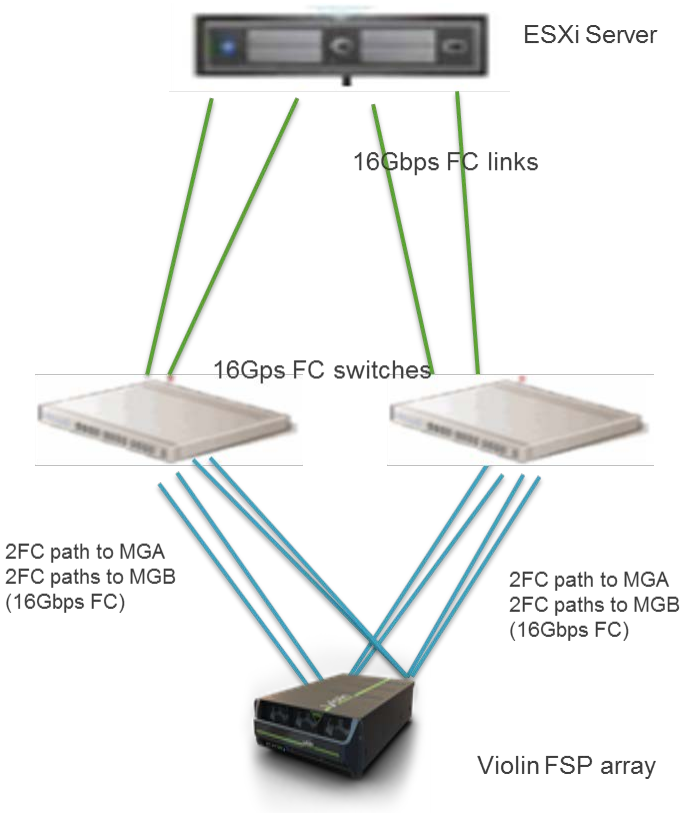
SAN Connectivity

Fibre Channel SAN Connectivity

To maintain a constant connection between ESXi hosts and storage, and to provide load balancing, VMware supports multipathing in ESXi environments. The recommended best practice in SAN design is to use a dual-fabric implementation with two separate SAN fabrics, which provides redundancy in the deployed SAN topology. This implementation requires more connections and a proper design to make sure that you have sufficient bandwidth across the connections for each fabric and that they are correctly paired together on the HA pairs at the back of the memory array.

This design provides primary and redundant paths from both SAN switches and distributes the traffic between FC ports associated with both memory gateways on the Violin array, allowing for the failure of either a SAN switch or a memory gateway without a loss of service. High availability is implemented by design between the ports provided by each memory gateway on the Array and will be made available for host connectivity when properly configured.

Violin Memory 7000 series FSP relies on the VMware Native Multipathing Plug-In for path redundancy in this configuration. This allows multiple streams from the host to be received on the two ports that makeup a high availability pair. Create the LUNs as type VMware, shared storage. Present all LUNs to each SAN Client in the VMware host cluster. Please follow the *Concerto 7000 All Flash Array Best Practices Guide* for connectivity best practices.



FSP FC Connectivity

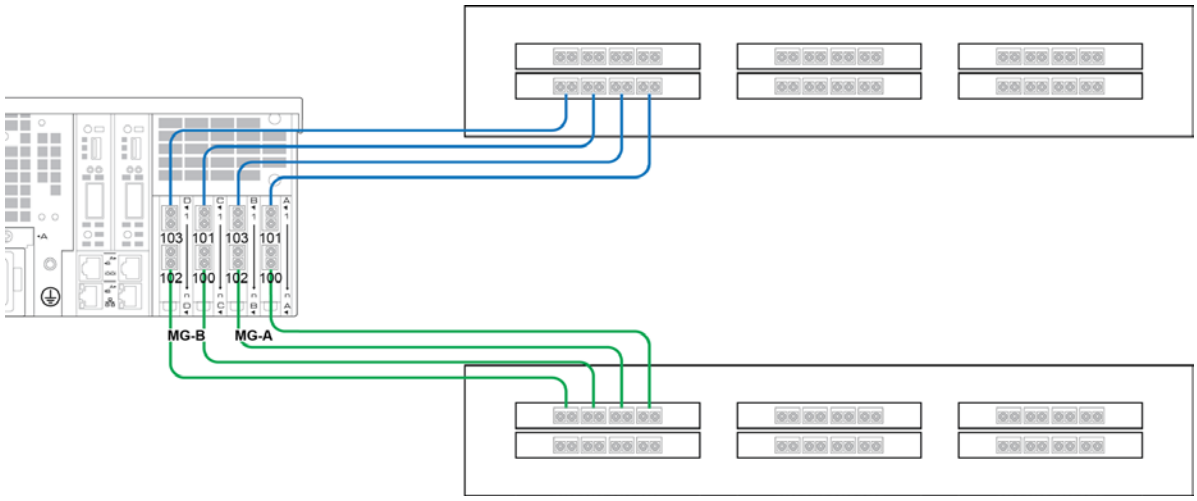
Fibre Channel Zoning Max Paths per LUN

Using the VMware vSphere® Web Client, check that each node in the cluster has visibility to all LUNs. Verify the proper number of paths from each host to each LUN. For example, when switch zoning is completed to the best practices, and each host has two Fibre Channel ports, there will be eight paths per LUN if using one switch, or four paths per LUN if using two switches for a highly available environment VMFS Datastore. Consider this recommendation when performing SAN design for multipathing configurations.

- Example with 2 server HBA ports per physical server:

  Fabric1: Zone 1 server HBA port with 4 of 8 Violin Target ports.

  Fabric2: Zone 1 server HBA port with 4 of 8 Violin Target ports.

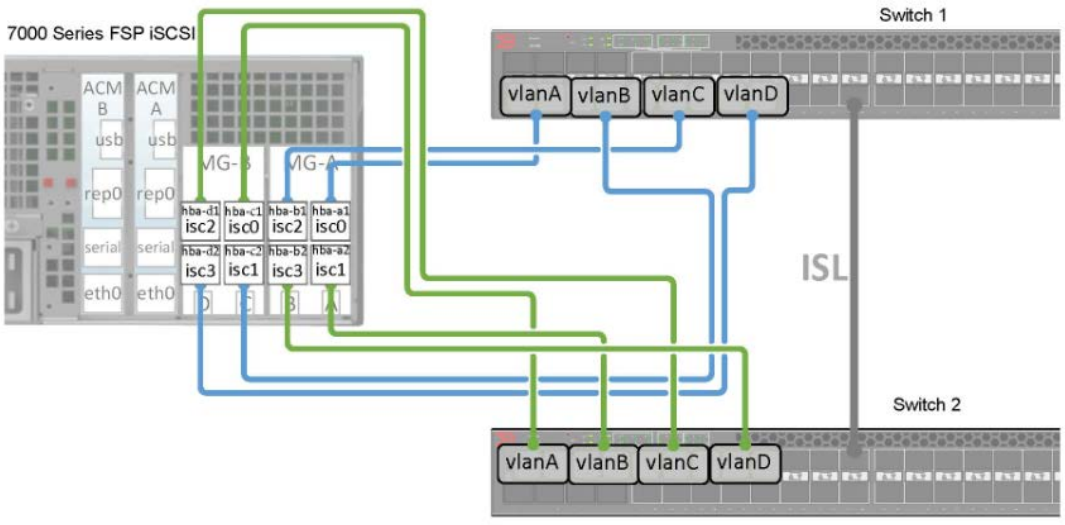


FSP FC Ports and Switch Connection

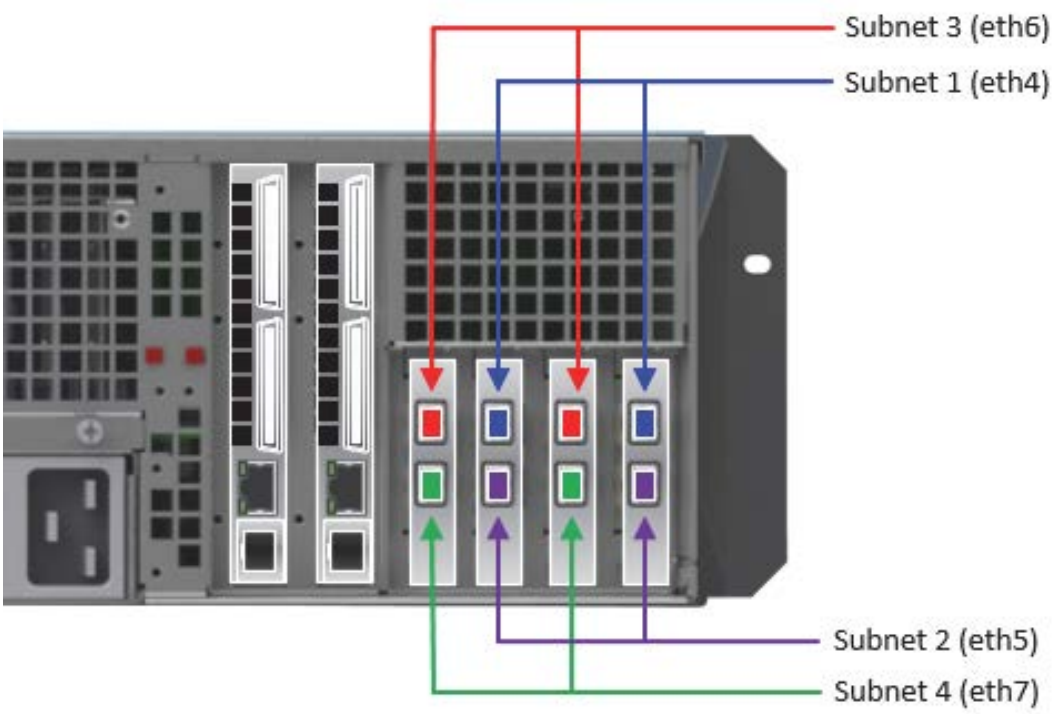
iSCSI SAN Connectivity

iSCSI is a popular storage system interconnect for Storage Area Networks (SANs). Benefits including cost effectiveness and high bandwidth over high speed network (10GbE, 40GbE, 100GbE). iSCSI runs over Ethernet networks; therefore, administrators can leverage the option of using existing infrastructure. Violin Memory Flash Storage Platform supports both FC and iSCSI implementations. Most of the best practices mentioned in this guide apply to both. The differences will be highlighted.

Violin Memory flash storage delivers high performance and high availability when there are multiple paths from the host server to the storage array. The best practice is to have separate SAN Fabrics, where there are two network switches in the SAN for high availability. If this is not possible, a single network switch topology is supported but offers less availability, as the switch is a single point of failure. With dual switches, a path failure can merely reduce the IOPS/bandwidth to the client until the path is restored. Figures below show a Violin Flash Array connected into an iSCSI SAN.



FSP iSCSI Ports and Switch Connection



Network configuration

The interface pair on the array for the subnet and the assigned NIC port on the ESXi host should all be configured with IP addresses in the same subnet. Interfaces sharing the same subnet are connected to each other and you can ping the adjacent interface. Example:

	ESXi Interface	Array Interface	MG-A	MG-B
Subnet 1	vmnic0	isc0	192.168.0.61	192.168.0.62
Subnet 2	vmnic1	isc1	192.168.1.61	192.168.1.62
Subnet 3	vmnic2	isc2	192.168.2.61	192.168.2.62
Subnet 4	Vmnic3	isc3	192.168.3.61	192.168.3.62

Network assignment

iSCSI Jumbo Frame

This feature can deliver additional throughput by increasing the size of the payload in each frame from a default MTU of 1,500 to an MTU of 9,000, Violin's recommendation. All devices sitting in the I/O path (iSCSI target, physical switches, network interface cards and VMkernel ports) must be able to implement jumbo frames for this option to provide the full benefits.

For example, if the MTU is not correctly set on the switches, the datastores might mount but I/O will fail. A common issue with jumbo-frame configurations is that the MTU value on the switch isn't set correctly. In most cases, this must be higher than that of the hosts and storage, which are set to 9,000. Switches must be set higher, to 9,198 or 9,216 for example, to account for IP overhead. Refer to switch-vendor documentation before attempting to configure jumbo frames.

To verify if Jumbo Frames (MTU of 9000) are enabled end to end, ssh into ESXi server, run the command:

```
# vmkping -d -s 8972 x.x.x.x
```

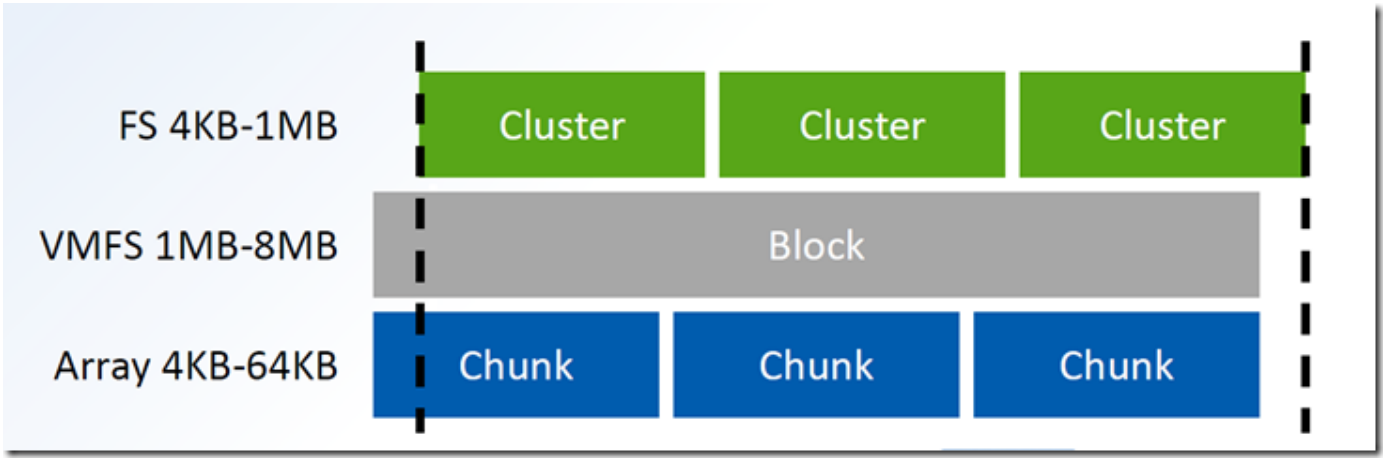
Where x.x.x.x is the array gateway address, and MTU size is 8972 (9000 - 8 bytes (ICMP header) - 20 bytes (IP header) for a total of 8972).

Tuning vSphere

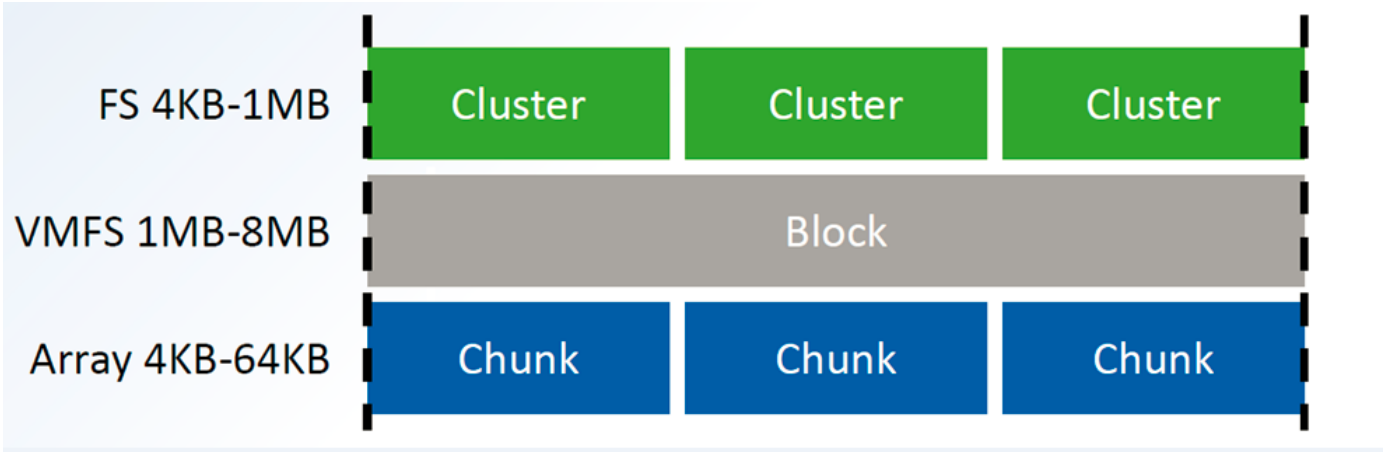
- GOS and VMFS 4K Alignment
- Partial IO
- Path Policy MPIO
- I/O Queue Depth Settings
- Fibre Channel Storage Queueing
- Sample Scripts
- iSCSI Storage Queueing
- ParaVirtual SCSI Adapter (PVSCSI)
- Modify PVSCSI Queue Depth
- Verify VAAI is enabled
- Increase XCOPY
- Eager-Zero Virtual Disk
- VMware Storage I/O Control
- Number of LUNs

GOS and VMFS Alignment (per VM)

A misaligned partition results in additional I/O; this incurs a penalty on latency and throughput. A correctly aligned partition eliminates the additional I/O and optimizes overall performance.



Guest OS Unaligned



Guest OS Aligned

Verify VMDK Guest Disk Partitions are 4K Aligned

- VMFS Datastore: Automatically Aligned when created from GUI
- Guest OS: Handled identically as Non-Virtualized OS
- Build OS Templates that are Pre-Aligned, avoiding duplication of effort
- Use EagerZeroThick disks from the beginning, where possible

---

***Note**: Modern Operating systems such as Windows Server 2012, 2008, Windows 7, RedHat Enterprise Linux/CentOS 7 automatically align their file system correctly by default. Creating VMFS datastores via vCenter 5 or later will automatically align VMFS blocks with the Violin array chunk size.*

If older versions of vSphere or Guest Operating Systems are being used or if the current environment had been upgraded, it is very important to verify that the VMFS and VM guest filesystems are aligned. There are different tools available from third parties and OS vendors that can verify and correct alignment.

How to Detect Mis-aligned on Windows Guests

One method is running this command (divide the StartingOffset with 4096 to verify if the offset is aligned with 4K):

```
C:\Windows\system32>wmic partition get Name, StartingOffset
Name                      StartingOffset
Disk #0, Partition #0     32256              NOT Aligned
Disk #0, Partition #1     65536              Aligned 4K
Disk #1, Partition #0     1048576            Aligned 4K
Disk #2, Partition #0     105906176          Aligned 4K
```

Another method is to run msinfo32 on the guest VM by selecting Start > All Programs > Accessories > System Tools > System Information. Next, navigate to Components > Storage > Disks and check the value for Partition Starting Offset. For misaligned VMs, you typically find that the VM is running with a default starting offset value of 32,256, which is not completely divisible by 4,096; hence, the partition is not aligned.

How to Detect Mis-aligned on Linux Guests

To detect the Linux filesystem starting offset within a VM, use tools:

- KPartEd/PartEd
- fdisk

Example with fdisk:

```
[root@db1 ~]# fdisk -l /dev/sdb
Disk /dev/sdb: 107.3 GB, 107374182400 bytes
255 heads, 63 sectors/track, 13054 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1               63       13054   104358240   83  Linux     NOT Aligned
    Device Boot      Start         End      Blocks   Id  System
/dev/sdb2               64       13054   104358240   83  Linux     Aligned 4K
```

Partial IO

A major consideration for Flash-based Memory Array system performance is 4-Kbyte address alignment on I/O requests. If a partition is not properly aligned, this will affect read/write I/O performance on the array. As stated earlier in this guide, VMFS partitions and most modern operating system filesystems are aligned at a 4KB automatically during installation. If not properly aligned, you will see high performance penalty due to large amount of Read-Modify-Write (RMW) operations.

Even if the partitions are aligned correctly, some applications such as MS SQL may have performance issues due to partial IO. Although data disks are formatted with 4k block size, the transaction logs are formatted with the physical block size presented by vSphere which is 512 bytes. If SQL asks for a sub-4k write to an existing transaction log file, then VMWare/Windows all pass along the sub-4k write. Then the array gets it and it's sub-4k so it has to do a RMW. This is because our array is formatted with

4k block size. Partial IOs, at high levels, can deteriorate array performance so it is recommended to keep them below 5% of total system IOPs.

For more information, please see https://kb.vmware.com/kb/2091600, https://kb.vmware.com/kb/2003813, https://kb.vmware.com/kb/1003565, https://kb.vmware.com/kb/2079071.

Path Policy - MPIO

For best performance and load balancing, set the multipath policy to Round-Robin (RR) for all Violin LUNs. This policy instructs the ESXi host to rotate all active paths. Round Robin will load balance across paths for different LUNs, and helps guarantee that all LUNs can perform past the connection limit. This should be done for both ALUA and non-ALUA LUNs.

Violin 7000 Storage Array Type Plug-in (SATP) rules are listed as VMW_SATP_ALUA. The default Path Selection Policy (PSP) for all devices claimed by VMW_SATP_ALUA is VMW_PSP_MRU. Therefore, the default needs to be changed to PSP to VMW_PSP_RR.

There are three ways to change the default PSP. One way is to set it on a per LUN basis. A reboot is not required in this case; the change takes effect immediately. However, the changes will not persist reboots.

Another way is global PSP change for all VMW_SATP_ALUA LUNs. This change requires a server reboot and will persist reboots. However, this will affect all other storage vendors using the default SATP on a given server.

Best practice is the third way, which sets the default PSP for Violin LUNs only and does not affect all other storage vendors. Although this change does not require a reboot, you need to un-claim and reclaim each device, and then do a storage rescan. Best practice is to simply reboot the server.

In addition, we recommend setting the IOs per path from default of 1000 to 4. Performance testing in our lab shows significant improvement in storage performance.

***Note**: Best Practice is to set the default policy for Violin LUNs only and I/O per path set to 4, and then reboot the server.*

Set Multipathing to round robin and IOs per path for Violin LUNs only

To check the existing path selection policy:

1. Log into the ESXi server, KB article 1019852.
2. Run this command to list the default PSP for Violin SATP

```
# esxcli storage nmp satp rule list | grep Violin
```

To change the default PSP for Violin LUNs only and set the preferred number of IOs per path to 4:

1. Run this command to change the pathing policy for VMW_SATP_ALUA for Violin:

```
# esxcli storage nmp satp rule add –s “VMW_SATP_ALUA” –V “VIOLIN” –M “CONCERTO ARRAY” –
P “VMW_PSP_RR” –O “iops=4”
```

2. Verify the pathing policy for Violin:

```
# esxcli storage nmp satp rule list | grep Violin
```

3. Reboot the ESXi host:

```
# reboot
```

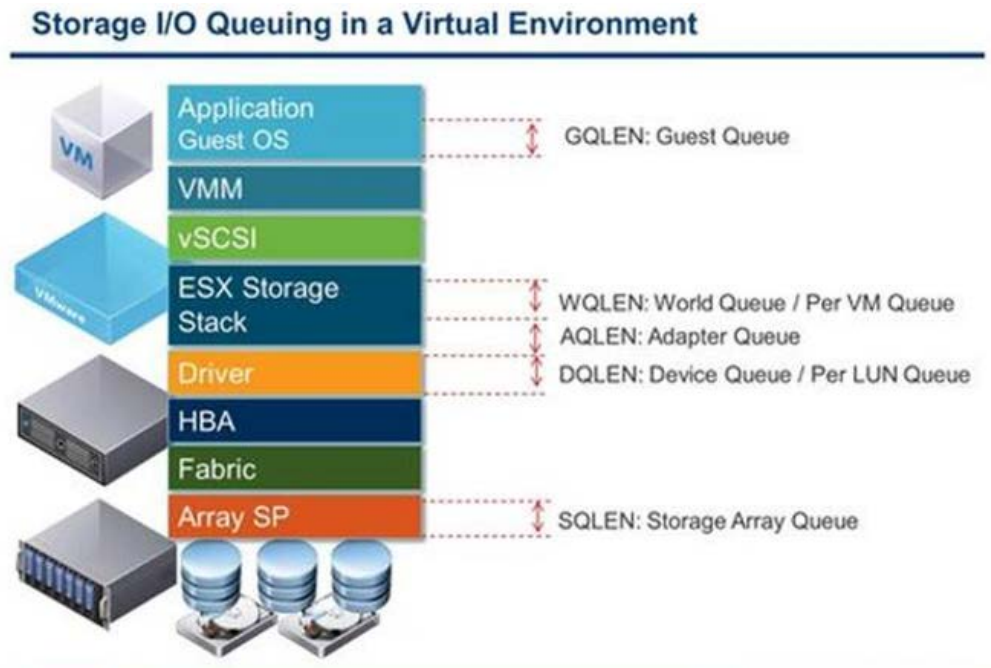
4. To display all devices multipath and iops per path policy

```
# esxcli storage nmp device list
```

If server reboot is not possible at this time, you can set the multipathing and iops per path policies by individual LUNs by following the scripts *set-rr.sh* and *set-iopsperpsp.sh* in the script section below. Because changing the policies by individual LUNs alone will not persist reboots, set the global setting above so that the policies will persist reboots.

I/O Queue Depth Settings

In this section, we will discuss the best practices for queues in the ESX storage stack in the diagram below. As you can see, there are many I/O queues in a virtual environment from the application and the guest OS down to the storage array. Although there are some advantages in tuning queue depths, if the bandwidth is too restrictive and I/Os are not spread across multiple paths as recommended in the previous sections, increasing the queue depths will only add on to the existing latency. For many applications, the default queue sizes are generally fine and do not require adjustments. But for use cases with high levels of consolidation or very intensive storage workloads, some of the vSphere queues may need to be adjusted for optimal performance. Below are the recommended settings for those use cases. To see the values of WQLEN, AQLEN, DQLEN, please see KB article 1027901. Please note, in this document, "LUN" and "Storage device" can be used interchangeably.



WQLEN is the queue length per VM. The default value is 32. This value represents the maximum number of I/Os one individual Virtual Machine can issue all the way down to the LUN. When there is only one VM issuing I/O to the LUN, then the LUN queue depth discussed below is used. When there are multiple VMs sharing the same LUN, then VM advanced setting *Disk.SchedNumReqOutstanding* throttles the I/O per VM to this setting. However, the per-host

parameter *Disk.SchedNumReqOutstanding* has been deprecated in vSphere 5.5. The setting is now per LUN. When PVSCSI is used in the VM (also discussed later), then queue depth per VM is the PVSCSI queue depth.

AQLEN is the queue length the HBA adapter is configured to support, e.g. 2176 for Qlogic HBA. It is a parameter passed from the adapter to the kernel and is not customizable.

DQLEN is the maximum number of outstanding vmkernel active commands to a LUN. This is sometime referred to as HBA LUN queue depth or simply LUN queue depth. This parameter should be changed from the default of 64 to take advantage of the high performance Violin storage arrays.

There are three ways to change the LUN queue depth. If a single VM is issuing I/O, the global HBA driver LUN queue depth is applicable. When multiple VMs are simultaneously issuing I/Os to a LUN, then per LUN maximum outstanding disk requests are used, and the global value is ignored. Another way LUN queue depth can be changed is by enabling Storage I/O Control (SIOC). If enabled, the per-LUN maximum outstanding disk requests will be ignored. It will adjust the device queue depth, but cannot go beyond the global setting. Best practice is to set the global value to 1024 and the per LUN values to 256. The following settings should be applied to all ESXi servers that have the LUN presented to them. More details can be found in KB articles 1267 and 1268.

***Note**: DQLEN in esxtop only shows the global HBA LUN queue depth setting.*

Fibre Channel Storage Queueing

To change the global HBA LUN queue depth setting from the default, first verify which HBA native driver is currently loaded:

1. For QLogic native drivers ESXi 5.5 and newer:

   Log into the ESXi server, KB article 1019852.

   ```
   # esxcli system module list | grep qln
   ```

   For Emulex:

   ```
   # esxcli system module list | grep lpfc
   ```

   For Cisco UCS:

   ```
   # esxcli system module list | grep fnic
   ```

2. To display the current HBA queue depth:

   For Qlogic:

   ```
   # esxcli system module parameters list -m qlnativefc | grep <parameter>
   ```

   Parameter:

   For vSphere 6.x use qlfxmaxqdepth

   For vSphere 5.x use ql2xmaxqdepth

For Cisco UCS:

```
# esxcli system module parameters list -m fnic | grep fnic_max_qdepth
```

3. To set the HBA Queue Depth to 1024:

   For Qlogic HBA:

   ```
   #esxcli system module parameters set -p qlfxmaxqdepth=1024 -m qlnativefc
   ```

   For vSphere 6.x use qlnxmaxqdepth

   For vSphere 5.x use ql2xmaxqdepth

   For Cisco UCS:

   ```
   #esxcli system module parameters set -p fnic_max_qdepth=1024 -m fnic
   ```

***Note**: Reboot your host. Verify changes have been made with the display commands above.*

Per LUN Queue Depth

The LUN queue depth can also be set per LUN maximum outstanding disk requests. Maximum value is 256. Please note, "LUN" and "device" can be used interchangeably.

1. To display LUN names:

   ```
   #esxcli storage vmfs extent list
   ```

2. To display queue depth per LUN:

   ```
   #esxcli storage core device list -d naa.xxx
   ```

3. To set LUN queue depths to 256:

   For each LUN set LUN queue depth (reboot not required)

   ```
   #esxcli storage core device set -d naa.xxx  -O  256
   ```

   Where: "naa.xxx" is the LUN identifier

***Note**: A reboot is not required for this setting.*

Sample scripts to set parameters per LUN

Note: *If the global settings for Violin LUNs have not been set, the individual LUN setting below will not persist reboots.*

To display Round Robin multipath policy for all Violin LUNs:

```
#cat display-rr.sh
#!/bin/sh
# display-rr:
for i in $(esxcli storage core device list | grep -i 'Violin Fibre Channel Disk' | awk '{print
$7}' | sed  's/[^a-zA-Z0-9.]/ /g'); do
esxcli storage nmp psp roundrobin deviceconfig get --device=$i
done
```

To set Round Robin multipath policy for all Violin LUNs:

```
#cat set-rr.sh
#!/bin/sh
# set-rr:
for i in $(esxcli storage core device list | grep -i 'Violin Fibre Channel Disk' | awk '{print
$7}' | sed  's/[^a-zA-Z0-9.]/ /g'); do
esxcli storage nmp device set --device=$i --psp=VMW_PSP_RR
done
```

To set iops per path policy for all Violin LUNs:

```
# cat set-iopsperpsp.sh
#!/bin/sh
# set-iopsperpsp:
for i in $(esxcli storage core device list | grep -i 'Violin Fibre Channel Disk' | awk '{print
$7}' | sed  's/[^a-zA-Z0-9.]/ /g'); do
echo doing $i
esxcli storage nmp psp roundrobin deviceconfig set --iops=4 --type=iops --device=$i
done
```

To set queue depth for all Violin LUNs:

```
# cat set-qdepth.sh
#!/bin/sh
# set-qdepth:
for i in $(esxcli storage core device list | grep -i 'Violin Fibre Channel Disk' | awk '{print
$7}' | sed  's/[^a-zA-Z0-9.]/ /g'); do
echo doing $i
esxcli storage core device set -d $i -O 256
done
```

iSCSI Storage Queuing

AQLEN is the queue length the HBA adapter is configured to support. Default 1024 for software iSCSI adapter.

The iSCSI adapter parameter can be changed.

esxcli system module parameters list -m iscsi_vmk |egrep -i "intr|qdepth“

esxcli system module parameters set -p "iscsivmk_HostQDepth=8192" -m iscsi_vmk

(Note: A reboot of the ESXi host is required.)

There are two ways to change the LUN queue depth. The same settings should be applied to all ESXi servers that have the LUN presented to them

Per LUN Queue Depth: When multiple VMs are simultaneously issuing I/Os to a LUN

esxcli system module parameters list -m iscsi_vmk |egrep -i "intr|qdepth“

esxcli system module parameters set -p "iscsivmk_LunQDepth=1024 iscsivmk_HostQDepth=8192 " -m iscsi_vmk

(Note: A reboot of the ESXi host is required.)

Storage I/O Control(SIOC): If enabled, the LUN queue depth will be ignored. It will adjust the queue depth, but cannot go beyond the LUN queue depth setting.

Paravirtual SCSI (PVSCSI) Adapters

Violin recommends the use of VMware Paravirtual high-performance storage adapters (PVSCSI) on guest virtual machines to take advantage of better CPU and storage performance with heavy I/O workloads. PVSCSI adapters can improve the throughput on virtual machines whose workloads drive very high I/O rates on low latency storage, such as Violin Flash Arrays. For example, each PVSCSI can do approx. 80K IOPS.

As a best practice, administrators should use multiple virtual SCSI controllers on a virtual machine as the number of virtual disks used in the solution grows. Spread the virtual disks across multiple controllers to take advantage of guest OS ability to submit I/Os to multiple disks concurrently. Note that VMware allows a maximum of four SCSI controllers to be assigned to a virtual machine at any time.

As long as the VM operating system is supported in KB 1010398, best practice is to use PVSCSI over default LSI. Performance tests have shown improvements between 10-70% depending on the application and environment. Information on how PVSCSI works is in KB 1017652. To configure VMware Paravirtual SCSI adapters on virtual machines, please follow the instructions in KB 1010398.

***Note**: As a best practice there should be multiple SCSI controllers and the disks should spread across controllers. Beginning with ESXi 5.0 there is a limit of four SCSI Controllers per VM.*

Modify PVSCSI Queue Depth

Just as there are queues in the vSphere storage stack, there are queues in the PVSCSI adapter. For non-I/O intensive workloads, the queue depth default values of 64 for device and 254 for adapter should work fine. However, large scale workloads with intensive I/O patterns require PVSCSI adapter queue depths greater than the default values, up to 256 for device and 1024 for adapter. Follow step 2 in the KB article 2053145 to modify the Paravirtual Queue Depths.

***Note**: It is strongly recommended that you back up Windows Registry prior to editing.*

Verify VMware VAAI is Enabled

The Violin 7000 Series Flash Storage Platforms implement VMware storage APIs Array Integration (VAAI) to offload certain functions from the VMware server to the FSP for enhanced performance. More details on VAAI can be found in the following white paper. The following VAAI optimizations are available in Concerto 7.5 and newer:

Advanced Parameter Name	Description
HardwareAcceleratedLocking	Atomic Test & Set (ATS), which is used during creation of files on the VMFS volume.
HardwareAcceleratedMove	Clone Blocks/Full Copy/XCOPY, which is used to copy data.
HardwareAcceleratedInit	Zero Blocks/Write Same, which is used to zero-out disk regions.
EnableBlockDelete*	SCSI UNMAP, which is used to reclaim space for thin provisioning LUNs

*For thin and dedup LUNs only.

***Note**: The above VAAI APIs will only optimize I/Os local to the Violin Memory Gateway.*

1. To verify support:

```
# esxcli storage vmfs extent list

# esxcli storage core device vaai status get -d naa.xxx
```

```
~ # esxcli storage core device vaai status get -d naa.6001b975db7b4b21ee39000056213164
naa.6001b975db7b4b21ee39000056213164
   VAAI Plugin Name:
   ATS Status: supported
   Clone Status: supported
   Zero Status: supported
   Delete Status: supported
~ #
```

2. To modify support:

   To enable or disable the VAAI primitives, use the following CLI commands:

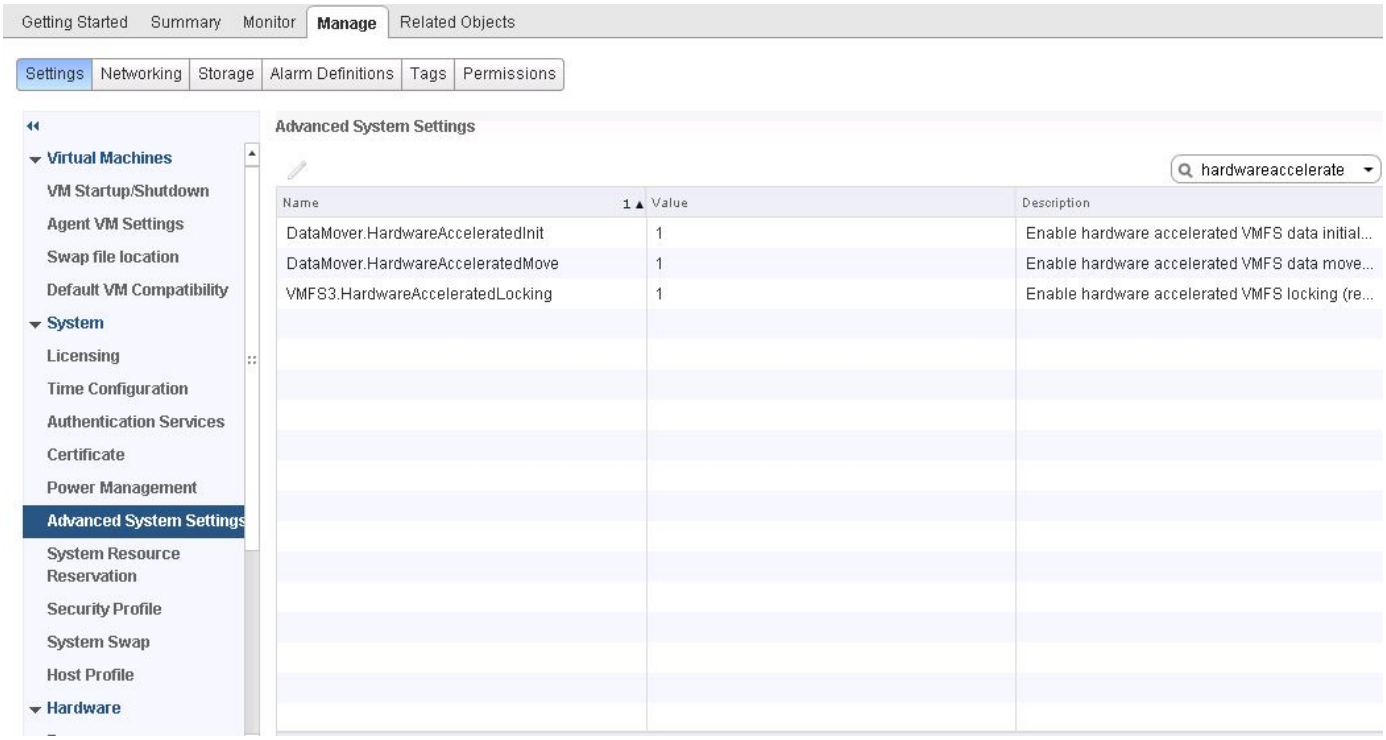
```
# esxcli system settings advanced set --int-value <x> --option
/VMFS3/HardwareAcceleratedLocking

# esxcli system settings advanced set --int-value <x> –option
/DataMover/HardwareAcceleratedMove
```

```
# esxcli system settings advanced set --int-value <x> –option
/DataMover/HardwareAcceleratedInit
```

Where: int-value <x> 0 is off, 1 is on

***Note**: For EnableBlockDelete, enabling/disabling occurs on the array and not on the ESXi host.*

The VAAI settings can also be viewed in vSphere Web Client under advanced setting in the ESXi host. See the figures below.



VAAI Plugin Advanced Settings: Acceleration

More information on enabling and disabling VAAI are available in KB article 1021976.

Increase the XCOPY Size

Violin recommends that the XCOPY size be changed from the default of 4MB to 16MB. Increasing the XCOPY size (MaxHWTransferSize) to 16MB will improve performance of cloning and storage vMotion by approximately 15%.

1. To see the current value on each server:

   ```
   # esxcli system settings advanced list -o /DataMover/MaxHWTransferSize
   ```

2. To set the value to 16MB on each server:

   ```
   # esxcli system settings advanced set -o /DataMover/MaxHWTransferSize -i 16384
   ```

Eager-Zeroed Virtual Disks

For high I/O workloads, Violin recommends the use of eager-zeroed thick virtual disks for best performance on virtual machines. Note that because zeroes are written to thick eager-zero disks upon creation, they take more time to create than thin or lazy zeroed disks.

For information on how to create thick eager-zeroed virtual disks for your virtual machines, see the VMware document *vSphere Virtual Machine Administration* on the VMware Support Web site.

VMware Storage I/O Control

Storage I/O Control (SIOC) is an ESXi cluster-wide performance isolation solution introduced by VMware in vSphere 4.1. SIOC protects virtual machines on a storage LUN from negative performance impact due to a single virtual machine with an I/O-heavy workload. The issue is more commonly known as the "noisy neighbor" problem.

Note: SIOC controls storage workloads by modifying the per-LUN queue depth, but SIOC cannot increase the queue depth beyond the configured global HBA LUN queue depth.

Violin recommends enabling Storage I/O Control in your vSphere environment only if you have VMs with different SLAs. It will increase or decrease the LUN queue depth based on the shares assigned to the VMs. Please note if SIOC is configured, the per-LUN device queue depth will be ignored.

Number of LUNs

For thick and thin LUNs, we recommend 2 or more LUNs. For dedup LUNs, we recommend 8 or more for best performance. This allows for parallelization through several layers including the operating system, hypervisor, filesystem, transportation system (fibre channel or iSCSI) and array.

MS SQL

Microsoft SQL is one of the most popular applications virtualized with vSphere. There are not many best practices settings needed for this application outside of the recommendations already mentioned in this document. Most settings are auto tuned by the OS or SQL Server. For best performance, we recommend separating workloads on multiple disks on different LUNs and controllers (PVSCSI).

- Data
- Log
- tempdb

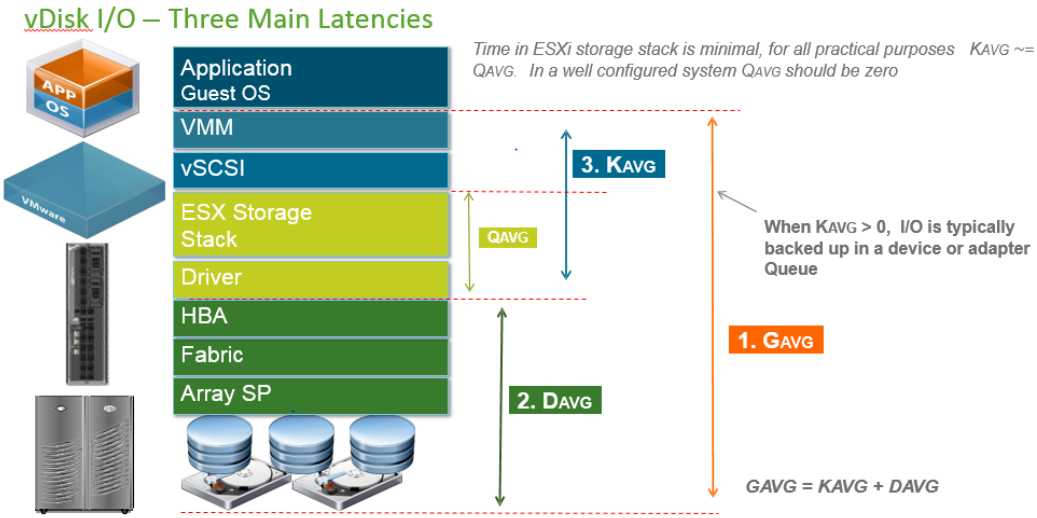
For more details, please see the *Violin Memory SQL Server Best Practices Guide.* Please also note the possible performance impact due to partial IO for SQL transaction logs as mentioned in the section above.

ORACLE Best Practices

For Oracle workload running on vSphere with Violin FSPs, please see the "Virtualized Oracle Considerations" section of the *Oracle Best Practices Guide*.

ESXi Troubleshooting Tips

For troubleshooting performance issues with storage, ensure best practices settings above have been applied first, then gather latency statistics using tools such as esxtop. Refer to KB article 1008205, and blogs Interpreting esxtop Statistics.



Source of I/O latencies

DAVG is a good indicator of performance of the backend storage. If I/O latencies are suspected to be causing performance problems, DAVG should be examined. Compare I/O latencies with corresponding data from the storage array. If they are close, check the array for misconfiguration or faults. If not, examine the data points in between the array and the ESXi server, e.g. FC switches, number of paths, bandwidth.

If KAVG is consistently high, then examine value listed under %USD, the percentage of queues actively being used. Sustained high values indicate the potential for queueing; you may need to adjust the HBA LUN queue depths. Queue sizes can be adjusted in a number of way as discussed in previous sections.